



Attorney Docket No. YOR920030395US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): L. D. Bergman et al.
Docket No.: YOR920030395US1
Serial No.: 10/697,752
Filing Date: October 30, 2003
Group: 2192
Examiner: Thuy Chan Dao

Title: Methods and Apparatus for Customizing
User-Interface Control in Existing Application

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter "Appellants") hereby appeal the final rejection of claims 29-37 of the above-identified application. The present application should be permitted to proceed to the Board for a decision on the merits.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, as evidenced by an assignment recorded on February 12, 2004 in the U.S. Patent and Trademark Office at Reel 014333, Frame 0076. The assignee, International Business Machines Corporation, is the real party in interest.

03/15/2010 MBELETE1 00000006 500510 10697752
01 FC:1402 540.00 DA

RELATED APPEALS AND INTERFERENCES

Appellants are not aware of any related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on October 30, 2003 with claims 1-36.

Claims 1-28 have been canceled without prejudice.

Claim 37 has been added.

Claims 29-37 are the pending claims.

Claims 29, 36 and 37 are the pending independent claims.

Claims 29, 31, 32 and 34-37 stand rejected under 35 U.S.C. § 102(b).

Claims 30 and 33 stand rejected under 35 U.S.C. § 103(a).

Claims 29-37 are appealed.

STATUS OF AMENDMENTS

There has been no amendment filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 29 recites apparatus for customizing a control of a user-interface of an existing application. The apparatus comprises a memory; and at least one processor, coupled to the memory operative to: (i) record a procedure description comprising a series of actions performed by a user in the application user-interface; and (ii) in response to the recording operation, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated.

An illustrative embodiment of an apparatus for customizing a control of a user-interface of an existing application is described in the specification at, for example, p. 10, l. 26 to p. 11, l. 16, with reference to FIG. 5. The apparatus comprising a memory and at least one processor coupled to the

memory (e.g., Specification, p. 11, ll. 14-24). The at least one processor operative to record a procedure description comprising a series of actions performed by a user in the application user-interface (e.g., Specification, p. 4, ll. 1-14); and in response to the recording operation, customize the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application (e.g., Specification, p. 5, ll. 14-25) for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated (e.g., Specification, p. 6, ll. 1-11).

Independent claim 36 recites an article of manufacture, stored in one or more memory devices, for customizing a control of a user-interface of an existing application comprising the steps of: recording a procedure description comprising a series of actions performed by a user in the application user-interface; and in response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated.

An illustrative embodiment of an article of manufacture, stored in one or more memory devices, for customizing a control of a user-interface of an existing application is described in the specification at, for example, p. 12, ll. 7-10, with reference to FIG. 5. Recording a procedure description comprising a series of actions performed by a user in the application user-interface (e.g., Specification, p. 4, ll. 1-14). In response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application (e.g., Specification, p. 5, ll. 14-25) for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated (e.g., Specification, p. 6, ll. 1-11).

Independent claim 37 recites a method for customizing a control of a user-interface of an existing application comprising the steps of: recording a procedure description comprising a series of actions performed by a user in the application user-interface; and in response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the

procedure description and generation of the series of actions performed by the user when the user-interface control is activated; wherein the recording and customizing steps are implemented via one or more software components that are loaded from a memory and executed by a processor device.

An illustrative embodiment of a method for customizing a control of a user-interface of an existing application is described in the specification at, for example, p. 12, ll. 7-10, with reference to FIG. 5. Recording a procedure description comprising a series of actions performed by a user in the application user-interface (e.g., Specification, p. 4, ll. 1-14). In response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application (e.g., Specification, p. 5, ll. 14-25) for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated (e.g., Specification, p. 6, ll. 1-11). One or more software components that are loaded from a memory and executed by a processor device (e.g., Specification, p. 11, ll. 14-24).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 29, 31, 32 and 34-37 stand rejected under 35 U.S.C. § 102(b) as being anticipated by S.A. Wolfman et al., “Mixed Initiative Interfaces for Learning Tasks: SMARTedit Talks Back” (hereinafter “Wolfman”).

II. Claim 30 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Wolfman in view of U.S. Patent No. 5,748,975 (hereinafter “Van De Vanter”).

III. Claim 33 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Wolfman in view of U.S. Patent Publication No. 2004/0078682 (hereinafter “Huang”).

ARGUMENT

Appellants respectively address grounds I, II and III presented above.

I. Anticipation of claims 29, 31, 32 and 34-37

A. Independent claims 29, 36 and 37

With regard to the § 102(b) rejection of independent claims 29, 36 and 37, Appellants note that “unless a reference discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it cannot be said to prove prior invention of the thing claimed and, thus, cannot anticipate under 35 U.S.C. § 102.” Net MoneyIN Inc. v. VeriSign Inc., 545 F.3d 1359, 88 U.S.P.Q.2d 1751, 1760 (Fed. Cir. 2008). Appellants assert that Wolfman fails to meet this standard.

Appellants contend that Wolfman fails to teach each and every limitation of the recited claims. Wolfman proposes an interface for machine learning that resembles a teacher-student (user-computer) relationship. Wolfman, Abstract. Wolfman describes a variety of rich interaction modes that enhance the learning process and presents a decision-theoretic framework, called DIAManD, for choosing the best interaction between the user (teacher) and computer (machine learning system). Id. The framework is applied specifically to SMARTedit and demonstrates experimental validation and preliminary user feedback. Id.

In contrast, by way of example, independent claim 29 recites apparatus for customizing a control of a user-interface of an existing application. The apparatus comprises a memory; and at least one processor, coupled to the memory operative to: (i) record a procedure description comprising a series of actions performed by a user in the application user-interface; and (ii) in response to the recording operation, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated. Independent claims 36 and 37 recite similar subject matter. As stated in the preamble of the independent claims, the present invention recites

customizing a control of a user-interface of an existing application. Wolfman does not propose to customize a control of a user-interface of an existing application as claimed.

Furthermore, Appellants respectfully submit that Wolfman fails to disclose the recited limitations in the body of the independent claim. For instance, Wolfman does not record a procedure description comprising a series of actions performed by a user in the application user-interface as recited in the independent claims and supported in the specification. An illustrative embodiment of the recording of a procedure description may be found in the specification of the present application at, for example, p. 4, ll. 1-7:

This invention comprises a software component that communicates with the operating system, capturing and delivering user events, such as keyboard clicks and mouse movements. This component can be thought of as a proxy, sitting between the operating system and the application. Procedures are recorded by capturing these low-level events as well as information about where application controls are located. This enables the recording component to make inferences about what components within the application have been activated.

Unlike the recited limitations, Wolfman describes techniques for programming by demonstration: “In contrast to regular macros, however, the PDB [programming by demonstration] system doesn’t merely record a series of keystrokes. Instead, it *generalizes* from one or more demonstrations to a robust, executable program.” Wolfman, col. 4, ll. 29-58. Therefore, Wolfman does not record a procedure description comprising a series of actions performed by a user in the application user-interface as recited in the claims; rather, Wolfman discloses a system for observing one or more user demonstrations and making generalizations about the one or more user demonstrations.

Next, Appellants submit that Wolfman does not teach or suggest, in response to the recording operation, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated. First, Wolfman does not teach customizing the control of the user-interface of the existing application by installing a user-interface control relating to the procedure description in the existing application. Customizing a control of a user-interface of an

existing application by installing a user-interface control is described in an illustrative embodiment of the specification at, for example, p. 8, ll. 15-18: “the request may be to add a new button to an existing toolbar within the application, to add a new toolbar containing a new button to the top-level application window, or to add a new item to an existing menu within the application interface.” See also Specification, p. 5, ll. 16-21 (creating and adding new controls or modifying existing controls). Second, by amending independent claims 29 and 36, Appellants have made clear that the step/operation of customizing the control of the user-interface by installing a user-interface control is performed in response to the recording step. This is not the case with Wolfman. The so-called customization of the user SMARTedit interface in FIG. 2 to the SMARTedit/DIAManD user interface of FIG. 4 that the Examiner points to is done by the application programmer not the application user. That is, the user gets the application the way it looks in FIG. 4; it is not customized by the user, i.e., in response to the user activated recording step. As the independent claims now recite, the customization of the interface, wherein a user-interface control is installed, is done in response to the recording step. This way, as explained at page 2, lines 2-4, of the present specification, the methodology allows for the installation of any new or modified control into any existing application to suit a specific user. The “modification” of the SMARTedit user interface in Wolfman is done by the programmer. The user of SMARTedit gets the application with the DIAManD control panel already in it.

The Examiner argues that Wolfman teaches customizing the control of the user-interface of the existing application (at Wolfman, col.4: 59- col.5: 25), in response to the recording operation (at Wolfman, col. 12, lines 8-23). See Final Office Action, dated 10/22/2009, p. 3. Specifically, the Examiner states at p. 3, ll. 17-20: “col. 12, lines 8-23, customizing an SMART edit user-interface by installing an “enhanced” SMART edit application, wherein said enhanced/customized user interface has an original user interface illustrated in col.5, FIG.2).” Appellants submit that the change in buttons between FIG. 2 and FIG. 4 is not the same as customizing a control of a user-interface of an existing application as recited in the claims and supported in the specification.

Appellants submit that the Wolfman reference is being mischaracterized. Appellants reiterate that Wolfman discloses techniques for machine learning. Wolfman describes how a user demonstrates a task by starting a macro recorder. Wolfman, col. 5, ll.18-22. After the user demonstrates a task, the

macro recorder is stopped and the computer predicts the user's next action. Wolfman, ll. 10-15. "The user interacts with the learning algorithm either by solving examples or by supervising the system's performance on an example." Wolfman, ll. 43-45. Appellants submit that Wolfman, FIGs. 2 and 4 depict interfaces with pre-programmed buttons to carry out this process. Wolfman is not customizing controls of an existing user-interface. For example, a process of adding (e.g., installing) a button onto one of the disclosed control panels is not disclosed by Wolfman.

Wolfman is concerned with presenting a way of training a computer to predict a task of a user. For instance, FIG. 4 shows three pre-programmed buttons, "Recording," "Bad Choice!," and "Done Recording." These buttons are used to train the computer to predict a user task. If the computer predicts a user task incorrectly, the user can interrupt and "rebuke" the computer by using the "Bad Choice!" button. Wolfman, col. 13, ll. 7-27. The computer may then make another prediction. Clearly, Wolfman's proposed process is not teaching the customization of a control of a user-interface of an existing application.

In addition to the above, by amending independent claims 29 and 36, Appellants have made clear that the user-interface control is specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated. This is not the case in Wolfman. That is, the "controls" in the "Interaction Control Panel" in the SMARTedit user interface of FIG. 4 of Wolfman are not specific to any recorded procedure. That is, the "controls" are generic, i.e., "record one step," "step," "run to end of example" . . . "jump and record an example."

It follows that Wolfman does not teach or suggest installing a user-interface control relating to the procedure description in the existing application. The Examiner argues that Wolfman teaches this limitation at Wolfman, col.4, l.59 to col.5, l. 25, FIG. 2, col. 12, FIG.4; and col. 13, ll. 5-21. See Final Office Action, dated 10/22/2009, p. 4.Appellants respectfully disagree. Wolfman at col. 13, ll. 22-27 states:

SMARTedit's user interface was altered to display the interaction choices as a set of radio buttons. DIAMandD's scores for the interactions are displayed by the contrast of the font and a horizontal gauge to the right of each interaction (as shown in Figure 4).

Finally, the “Bad Choice” button was added to allow users to rebuke DIAMandD’s choice.

Appellants note that this statement does not teach or suggest installing a user-interface control relating to the procedure description in the existing application. Wolfman is simply disclosing that Wolfman et al. altered the SMARTedit’s user interface by adding radio buttons to display the interaction choices. To consider this a teaching of installing a user-interface control relating to the procedure description in the existing application as recited in the claims is a mischaracterization of Wolfman. Further, Appellants note that the Examiner is parsing the claim language to present his argument. It is clear that the portions of Wolfman cited by the Examiner, when read together, fail to disclose “customizing the control of the user-interface of the existing application by installing a user-interface control relating to the procedure description in the existing application,” as recited in the claims and supported in the specification.

Further, Wolfman fails to teach or suggest customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated. Wolfman discloses the prediction of a user action rather than the generation of the series of actions performed by the user as recited in the claims. The generation of the series of actions is described in an illustrative embodiment of the specification at, for example, p. 6, ll. 4-11 (“These events simulate what the user would have done if she had activated the controls that were recorded during the recording phase. In effect, command player 112 is replaying the recorded procedure.”). Appellants submit that predicting a user’s actions is not the same as generating the series of actions performed by the user (e.g., replaying a recorded procedure). In fact, Wolfman associates a predicted user action with a probability of correction (Wolfman, col. 11, ll. 19-28), which confirms that Wolfman is not generating the series of actions performed by the user when the user-interface control is activated, instead, Wolfman is making a computer guess at the user’s next action wherein each guess may be confirmed by the user. See Wolfman, col. 13, ll. 7-22.

In addition to the above, even if the Examiner were correct in believing that Wolfman discloses each of the claimed limitations individually (which, for the reasons given above, he is not), Wolfman fails to teach all of the claimed limitations arranged or combined in the same way as recited in the claims. Appellants note that the claims recite recording a procedure description comprising a series of actions performed by a user in the application user-interface; and in response to the recording operation, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated. Wolfman does not disclose the recording of a procedure description followed by customizing the control of the user-interface, in response to the recording operation, of the existing application by installing a user-interface control specific to the procedure description in the existing application (i.e., Wolfman is not recording a procedure description and then installing a user-interface control in an existing application to carry out the recorded procedure description). Wolfman teaches a user demonstrating a task and the user using pre-installed buttons to teach a computer to predict the user's task via trial and error.

For at least these reasons, Appellants assert that independent claims 29, 36 and 37 are not anticipated by Wolfman.

B. Dependent claims 31, 32, 34, and 35

It follows that claims 31, 32, 34, and 35 are not anticipated by Wolfman at least by virtue of their respective dependencies from independent claim 29. Further, dependent claims 31, 32, 34, and 35 recite patentable subject matter in their own right, and Appellants respectfully submit that the portions of Wolfman cited by the Examiner do not teach the limitations of the dependent claims. For instance, Wolfman at col. 5, ll. 18-26 does not teach or suggest registering the procedure capturer with the operating system to receive notification of user actions and system actions as recited in claim 31 and supported in the specification. An illustrative embodiment of registering may be found in the specification at, for example, p. 7, ll. 1-6. The text cited by the Examiner discloses the use of a macro recorder. The Examiner argues that Wolfman's macro recorder was inherently registered with the

operating system. “In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art.” Ex parte Levy, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original). However, the examiner did not provide objective evidence or cogent technical reasoning to support the conclusion of inherency.

Further, Wolfman does not teach or suggest the specific steps of recited claim 32. Dependent claim 32 recites the steps of: receiving notification of user action within the application interface at the procedure capturer; determining a result activated by user action at the procedure capturer; mapping the activated control into a generic description at the procedure capturer; adding the generic description to a procedure representation at the procedure capturer; determining if there are more user actions; and storing the procedure representation in the procedure capturer if there are no more user actions. The Examiner argues that Wolfman teaches these steps at Wolfman, col. 4, l. 60 to col. 5, l. 26. The cited portions of Wolfman clearly do not describe the recited steps of claim 32. For instance, Wolfman does not determine a result activated by user action at the procedure capturer, nor does Wolfman disclose mapping the activated control into a generic description at the procedure capturer. An illustrative embodiment of the steps of claim 32 may be found in the specification at, for example, p. 7, ll. 7-21.

Moreover, Wolfman does not teach or suggest that the at least one processor is further operative to execute an installed control as recited in claim 34. The Examiner argues that Wolfman teaches this limitation at Wolfman, col. 12, ll. 8-23. Appellants reiterate that the radio buttons of the altered SMARTedit interface disclosed by Wolfman are not installed controls as recited in the claims; therefore, Wolfman does not disclose executing an installed control.

Finally, Wolfman does not disclose altering the appearance of at least one existing user interface control as recited in claim 35. Altering the appearance of at least one existing user interface control is described in one illustrative embodiment of the specification at, for example, p. 9, l. 26 to p. 10, l. 9:

In addition to adding new controls to an existing application, the inventive technique described herein can be used to alter the visible appearance of existing controls within an application. This kind of alteration of appearance is commonly known as “skinning.” A set of alternate controls for existing operations can be created for any application. Underlying application controls may be invoked for the controls that are being reskinned, by simulating the actions (e.g. Mouse click, key press) required to activate that control. Thus, it is possible to have a control that does not use a prerecorded procedure from the procedure repository, but simply passes the events to the operating system and application. For example, an overlay window may have a round button where the original application had a square button. When the round button is pressed by the user, the proxy simulates a press of the original square button.

The Examiner argues that Wolfman teaches claim 35 at Wolfman, FIG. 2, FIG. 4, col. 5, and col. 12. Wolfman is not altering the appearance of at least one existing user interface control. If anything, Wolfman is completely changing the controls of a user interface. This is not the same as altering the appearance of an existing user interface control.

For at least these reasons, Appellants submit that Wolfman fails to anticipate dependent claims 31, 32, 34, and 35. Accordingly, Appellants respectfully request withdrawal of the § 102 rejection.

In view of the above, Appellants believe that claims 31, 32, 34, and 35 are in condition for allowance and respectfully request withdrawal of the § 102(b) rejection.

II. Obviousness of claim 30

Claim 30 is rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Wolfman in view of Van De Vanter. In particular, the Examiner acknowledges that Wolfman does not explicitly disclose the operation of recording a procedure description performed by a user in the application user-interface comprises the steps of transmitting at least one request for application user-interface structure information from a procedure capturer to an operating system and receiving the application user-interface structure information from the operating system at the procedure capturer, but asserts that Van De Vanter discloses this limitation (FIG. 1, FIG. 2; col. 7, l. 25 to col. 8, l. 11; col. 8, ll. 33-67; col. 9, ll. 8-59;).

However, Van De Vanter clearly does not teach or suggest wherein an operation of recording a procedure description performed by a user in the application user-interface comprising the steps of:

transmitting at least one request for application user-interface structure information from a procedure capturer to an operating system; and receiving the application user-interface structure information from the operating system at the procedure capturer. Nor is it clear how or why one would look to an on-the-fly typographical display as disclosed in Van De Vanter to improve an interface for machine learning that resembles a teacher-student relationship as disclosed in Wolfman.

The rejection of claim 30 under section 103 in view of Wolfman and Van De Vanter, alone or in combination, is therefore believed to be improper and should be withdrawn.

III. Obviousness of claim 33

Claim 33 is rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Wolfman in view of Huang. In particular, the Examiner acknowledges that Wolfman does not explicitly disclose the operation of installing the control relating to the procedure description in the existing application comprises the steps of sending a request to an operating system from a control installer to install the control; creating and installing the control in the application user-interface through the operating system; and registering a callback at the operating system using a location supplied by the control installer, but asserts that Huang discloses this limitation (paragraphs [0005], [0007], [0009], [0012], [0019], and [0050]).

Contrary to the Examiner's assertion, Appellants could find *no* disclosure or suggestion in Huang of operation of installing the control relating to the procedure description in the existing application comprises the steps of sending a request to an operating system from a control installer to install the control; creating and installing the control in the application user-interface through the operating system; and registering a callback at the operating system using a location supplied by the control installer. Claim 33 contains this limitation. It follows that Huang fails to remedy the deficiencies of Wolfman.

Thus, Wolfman and Huang, alone or in combination, do not disclose or suggest the operation of installing the control relating to the procedure description in the existing application comprises the steps of sending a request to an operating system from a control installer to install the control; creating and installing the control in the application user-interface through the operating system; and

registering a callback at the operating system using a location supplied by the control installer, as required by claim 33.

The rejection of claim 33 under section 103 in view of Wolfman and Huang, alone or in combination, is therefore believed to be improper and should be withdrawn.

Conclusion

Accordingly, Appellants respectfully request withdrawal of the respective §102 and §103 rejections of claims 29-37.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully submitted,

/William E. Lewis/

Date: March 10, 2010

William E. Lewis
Attorney for Applicant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

CLAIMS APPENDIX

29. Apparatus for customizing a control of a user-interface of an existing application comprising:

a memory; and

at least one processor, coupled to the memory operative to: (i) record a procedure description comprising a series of actions performed by a user in the application user-interface; and (ii) in response to the recording operation, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated.

30. The apparatus of claim 29, wherein the operation of recording a procedure description performed by a user in the application user-interface comprises the steps of:

transmitting at least one request for application user-interface structure information from a procedure capturer to an operating system; and

receiving the application user-interface structure information from the operating system at the procedure capturer.

31. The apparatus of claim 29, wherein the operation of recording a procedure description performed by the user in the application user-interface comprises the step of registering the procedure capturer with the operating system to receive notification of user actions and system actions.

32. The apparatus of claim 29, wherein the operation of recording a procedure description performed by the user in the application user-interface comprises the steps of:

receiving notification of user action within the application interface at the procedure capturer;

determining a result activated by user action at the procedure capturer;

mapping the activated control into a generic description at the procedure capturer;

adding the generic description to a procedure representation at the procedure capturer;

determining if there are more user actions; and
storing the procedure representation in the procedure capturer if there are no more user actions.

33. The apparatus of claim 29, wherein the operation of installing the control relating to the procedure description in the existing application comprises the steps of:

 sending a request to an operating system from a control installer to install the control;
 creating and installing the control in the application user-interface through the operating system; and
 registering a callback at the operating system using a location supplied by the control installer.

34. The apparatus of claim 29, wherein the at least one processor is further operative to execute an installed control.

35. The apparatus of claim 29, wherein the operation of installing the user-interface control comprises the step of altering the appearance of at least one existing user interface control.

36. An article of manufacture, stored in one or more memory devices, for customizing a control of a user-interface of an existing application comprising the steps of:

 recording a procedure description comprising a series of actions performed by a user in the application user-interface; and

 in response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated.

37. A method for customizing a control of a user-interface of an existing application comprising the steps of:

recording a procedure description comprising a series of actions performed by a user in the application user-interface; and

in response to the recording step, customizing the control of the user-interface of the existing application by installing a user-interface control specific to the procedure description in the existing application for automatic execution of the procedure description and generation of the series of actions performed by the user when the user-interface control is activated;

wherein the recoding and customizing steps are implemented via one or more software components that are loaded from a memory and executed by a processor device.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.